S10 – L3 - Malware Analysis – La memoria ed il linguaggio Assembly

Traccia

Nella lezione teorica del mattino, abbiamo visto i fondamenti del linguaggio Assembly.

Dato il codice in Assembly per la CPU x86 allegato qui di seguito, identificare lo scopo di ogni istruzione, inserendo una descrizione per ogni riga di codice. Ricordate che i numeri nel formato 0xYY sono numeri esadecimali.

Per convertirli in numeri decimali utilizzate pure un convertitore online, oppure la calcolatrice del vostro computer (per programmatori).

- 0x00001141 <+8>: mov EAX,0x20
- 0x00001148 <+15>: mov EDX,0x38
- 0x00001155 <+28>: add EAX,EDX
- 0x00001157 <+30>: mov EBP, EAX
- 0x0000115a <+33>: cmp EBP,0xa
- 0x0000115e <+37>: jge 0x1176 <main+61>
- 0x0000116a <+49>: mov eax,0x0
- 0x0000116f <+54>: call 0x1030 <printf@plt>

Svolgimento

Innanzitutto, identifichiamo le varie parti delle istruzioni:

- **0x00001141 <+8>: mov EAX,0x20**
  - **0x00001141**: Questo è l'indirizzo di memoria in cui si trova l'istruzione di assembly. Ogni istruzione nel programma ha un indirizzo univoco che consente alla CPU di sapere dove trovare l'istruzione successiva da eseguire.
  - **<+8>**: Questo rappresenta l'offset relativo rispetto all'inizio di una funzione o di un blocco di codice. Indica che questa istruzione si trova a 8 byte di distanza dall'inizio della funzione o del blocco a cui si riferisce. Questo offset è utile per gli sviluppatori o per i debugger per comprendere la posizione dell'istruzione all'interno del codice.
  - **mov**: Questa istruzione indica al sistema di copiare un valore
  - **EAX**: Indica il registro di destinazione a cui assegnare il valore
  - **0x20**: è il valore da copiare e da inserire nel registro, in formato esadecimale

    - **Descrizione:** Copia il valore esadecimale 0x20 (32 in decimale) nel registro EAX.
- **0x00001148 <+15>: mov EDX,0x38**
    - **0x00001148**: Questo è l'indirizzo di memoria in cui si trova l'istruzione di assembly.
    - **<+15>**: Indica che questa istruzione si trova a 15 byte di distanza dall'inizio della funzione.
    - **mov**: Questa istruzione indica al sistema di copiare un valore.
    - **EDX**: Indica il registro di destinazione a cui assegnare il valore.
    - **0x38**: è il valore da copiare e da inserire nel registro, in formato esadecimale.
    - **Descrizione**: Copia il valore esadecimale 0x38 (56 in decimale) nel registro EDX.
- **00001155 <+28>: add EAX, EDX**
    - **0x00001155**: Questo è l'indirizzo di memoria in cui si trova l'istruzione di assembly.
    - **<+28>**: Indica che questa istruzione si trova a 28 byte di distanza dall'inizio della funzione.
    - **add**: Questa istruzione indica al sistema di sommare i valori dei registri specificati.
    - **EAX, EDX**: I registri il cui contenuto sarà sommato.
    - **Descrizione**: Somma i valori contenuti nei registri EAX e EDX, e memorizza il risultato in EAX.
- **0x00001157 <+30>: mov EBP, EAX**
    - **0x00001157**: Questo è l'indirizzo di memoria in cui si trova l'istruzione di assembly.
    - **<+30>**: Indica che questa istruzione si trova a 30 byte di distanza dall'inizio della funzione.
    - **mov**: Questa istruzione indica al sistema di copiare un valore.
    - EBP, EAX: EAX è il registro sorgente da cui il valore sarà copiato in EBP, il registro di destinazione.
    - **Descrizione**: Trasferisce il valore corrente del registro EAX nel registro EBP.
- **0x0000115a <+33>: cmp EBP,0xa**
    - **0x0000115a**: Questo è l'indirizzo di memoria in cui si trova l'istruzione di assembly.
    - **<+33>**: Indica che questa istruzione si trova a 33 byte di distanza dall'inizio della funzione.
    - **cmp**: Questa istruzione indica al sistema di confrontare i valori specificati.
    - **EBP**: Il registro il cui valore sarà confrontato.
    - **0xa**: è il valore con cui il registro EBP viene confrontato, 10 in decimale.

    - **Descrizione**: Confronta il valore contenuto nel registro EBP con il numero decimale 10 (0xa in esadecimale).
- **0x0000115e <+37>: jge 0x1176 <main+61>**
    - **0x0000115e**: Questo è l'indirizzo di memoria in cui si trova l'istruzione di assembly.
    - **<+37>**: Indica che questa istruzione si trova a 37 byte di distanza dall'inizio della funzione.
    - **jge**: Questa istruzione indica al sistema di effettuare un salto condizionale se la condizione è vera.
    - **0x1176 <main+61>**: è l'indirizzo di destinazione del salto.
    - **Descrizione**: Esegue un salto all'indirizzo 0x1176 se il valore in EBP è maggiore o uguale a 10.